

ITW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Group Art Unit: 2186
Examiner: Unassigned

In re PATENT APPLICATION of

Applicant : Junichi TAMURA)

Appl. No. : 10/733,302)

Filed : December 12, 2003)

For : DATA REARRANGEMENT METHOD)

Atty. Dkt. : OKI 402)

**SUBMISSION OF
PRIORITY DOCUMENT**

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Submitted herewith is a certified copy of applicant's first-filed Japanese Application No. 2003-318493 filed September 10, 2003, the rights of priority of which were claimed on December 12, 2003 with the filing of the above-referenced application.

It is respectfully requested that receipt of this priority document be acknowledged.

Respectfully submitted,

Steven M. Rabin - Reg. No. 29,102
RABIN & BERDO, P.C.
Telephone: 202 371 8976
Telefax: 202 408 0924
CUSTOMER NO. 23995

November 15, 2004
Date

SMR:pjl

FEE ENCLOSED: \$
Please charge any further
fee to our Deposit Account
No. 18-0002

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 9 月 1 0 日
Date of Application:

出 願 番 号 特 願 2 0 0 3 - 3 1 8 4 9 3
Application Number:
[ST. 10/C] : [J P 2 0 0 3 - 3 1 8 4 9 3]

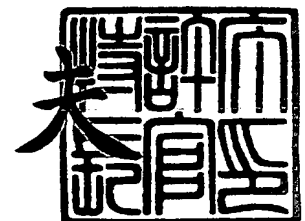
出 願 人 沖 電 気 工 業 株 式 有 限 公 司
Applicant(s):

CERTIFIED COPY OF
PRIORITY DOCUMENT

2 0 0 3 年 1 2 月 5 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願
【整理番号】 KT000438
【提出日】 平成15年 9月10日
【あて先】 特許庁長官 今井 康夫 殿
【国際特許分類】 G06F 7/24
【発明者】
 【住所又は居所】 東京都港区虎ノ門1丁目7番12号 沖電気工業株式会社内
 【氏名】 田村 純一
【特許出願人】
 【識別番号】 000000295
 【氏名又は名称】 沖電気工業株式会社
【代理人】
 【識別番号】 100095957
 【弁理士】
 【氏名又は名称】 亀谷 美明
 【電話番号】 03-5919-3808
【選任した代理人】
 【識別番号】 100096389
 【弁理士】
 【氏名又は名称】 金本 哲男
 【電話番号】 03-3226-6631
【選任した代理人】
 【識別番号】 100101557
 【弁理士】
 【氏名又は名称】 萩原 康司
 【電話番号】 03-3226-6631
【手数料の表示】
 【予納台帳番号】 040224
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1
 【包括委任状番号】 9707549
 【包括委任状番号】 9707550
 【包括委任状番号】 9707551

【書類名】 特許請求の範囲**【請求項 1】**

- a) 第 1 のデータ格納部にデータを格納するステップと、
- b) スタックにデータ並べ替え情報を格納するステップと、
- c) 前記第 1 のデータ格納部に格納された前記データを読み出し、該データを前記スタックに格納された前記データ並び替え情報に基づいて第 2 のデータ格納部に格納するステップとを有することを特徴とするデータ並び替え方法。

【請求項 2】

前記データ並び替え情報は、前記第 2 のデータ格納部のアドレスで構成されることを特徴とする請求項 1 記載のデータ並び替え方法。

【請求項 3】

前記第 1 のデータ格納部は、レジスタであり、

前記第 2 のデータ格納部は、ランダムアクセスメモリであることを特徴とする請求項 2 記載のデータ並び替え方法。

【請求項 4】

- a) 第 1 のデータ格納部に複数のデータを格納するステップと、
- b) スタックにデータ並べ替え情報を格納するステップと、
- c) 前記第 1 のデータ格納部に格納された前記複数のデータ各々を前記スタックに格納された前記データ並び替え情報に基づく順序で読み出し、該データを第 2 のデータ格納部に格納するステップとを有することを特徴とするデータ並び替え方法。

【請求項 5】

前記データ並び替え情報は、前記第 1 のデータ格納部のアドレスで構成されることを特徴とする請求項 4 記載のデータ並び替え方法。

【請求項 6】

前記第 1 のデータ格納部は、ランダムアクセスメモリであり、

前記第 2 のデータ格納部は、レジスタであることを特徴とする請求項 5 記載のデータ並び替え方法。

【請求項 7】

前記第 1 のデータ格納部及び前記第 2 のデータ格納部は、ランダムアクセスメモリであることを特徴とする請求項 5 記載のデータ並び替え方法。

【請求項 8】

- a) 第 1 のデータ格納部に複数のデータを格納するステップと、
- b) スタックにデータ並べ替え情報を格納するステップと、
- c) 前記第 1 のデータ格納部に格納された前記複数のデータの各々を読み出し、前記スタックに格納された前記データ並び替え情報に基づいて第 2 のデータ格納部に格納するステップとを有することを特徴とするデータ並び替え方法。

【請求項 9】

前記データ並び替え情報は、前記第 2 のデータ格納部のアドレスで構成されることを特徴とする請求項 8 記載のデータ並び替え方法。

【請求項 10】

前記第 1 のデータ格納部及び前記第 2 のデータ格納部は、ランダムアクセスメモリであることを特徴とする請求項 9 記載のデータ並び替え方法。

【請求項 11】

アドレス変換テーブルと、それに対応するスタックポインタとを用いて、前記読み出しと前記格納とを行うことを特徴とする請求項 1, 4, または 8 のうちのいずれか 1 項記載のデータ並び替え方法。

【請求項 12】

読み出したアドレスとオフセットレジスタとの論理和または加算を演算するステップをさらに含むことを特徴とする請求項 1, 4, または 8 のうちのいずれか 1 項記載のデータ並び替え方法。

【請求項 1 3】

前記スタックポインタの代わりに専用レジスタを用いることを特徴とする請求項 1 1 記載のデータ並び替え方法。

【請求項 1 4】

前記アドレス変換テーブルに格納するデータに、バイトライト情報を含めることを特徴とする請求項 1 1 記載のデータ並び替え方法。

【書類名】明細書

【発明の名称】データ並び替え方法

【技術分野】

【0001】

本発明は、デジタル通信やデジタル信号処理におけるデータ処理方法に関し、特に、データ並び替え方法に関するものである。

【背景技術】

【0002】

デジタル通信やデジタル信号処理の分野では、インターリーブ処理が行われる。インターリーブ処理とは、複数のグループに分割したデータを特定の規則に従って並び替えて、読み・書き・送受信などの処理を行うものである。

【0003】

例えば、Digital Signal Processor（デジタル信号処理を行うための演算処理装置。以下、「DSP」という）におけるインターリーブ処理では、DSPが演算処理するデータの順番は、メモリに書き込まれたデータの順番とは異なることがある。そして、DSPで使用するデータを演算処理順に並び替えるためには、多くの処理時間、多くの記憶容量を必要とする。

【0004】

データを並び替えるための従来の方法には、アドレス変換テーブルを用いるものがある。ここでは説明の簡略化のため、一次元の並び替えについて説明する。図8は、アドレス変換テーブルを用いた従来のデータ並び替え方法を説明するための、データ並び替え前後のデータ配置状態と、アドレス変換テーブルの状態を示す図である。図8において、「アドレス：0x0000」などはメモリの16進数表示によるアドレス番地を示し、A～Eは、メモリに格納され、並び替えが実行されて演算処理される、所定長さのデータを示す。

【0005】

図8に示すように、メモリには、アドレス（0x0000番地）からアドレス（0x0004番地）まで、5つのデータがA、B、C、D、Eの順に格納されている。そして、アドレス（0x2000番地）からアドレス（0x2004番地）までの「アドレス変換テーブル」には、データ並び替えの順が格納されている。このアドレス変換テーブルに格納された並び替え規則（（1）～（5））に従えば、データA、B、C、D、Eは、D、B、E、A、Cの順に、アドレス（0x1000番地）からアドレス（0x1004番地）の格納場所に並び替えられる（図8）。

【0006】

図9は、この従来の並び替え方法を実現するための流れ図と、プログラム例である。ここでは、3つのアドレス格納用のレジスタr0、r1、r2と、1つのデータ格納用レジスタa0を用いている。プログラム例において、「mov」は、「**（**」の左の値を「**）**」の右の値にコピーする命令であり、「**（**」付きの文字／数は、そのアドレスやレジスタに格納されるデータの値を、「**（**」無しの文字／数は、そのアドレス番地やレジスタそのものを示している。また、文字／数の右の「+」は、値を+1だけインクリメントすることを示している。

【0007】

図9において、まずS1で、アドレス変換テーブルの先頭アドレス（0x2000番地）をr0レジスタにセットする（S1）。次に、S2で、並び替え前のデータが格納される先頭アドレス（0x0000番地）を、r1レジスタにセットする（S2）。次に、S3で、r1レジスタに格納されたメモリ（I）のアドレス（0x0000）に、格納されているデータAを、a0レジスタにセットし、r1レジスタの値をインクリメント（+1）する（S3）。次に、S4で、r0レジスタのアドレスに格納されるアドレス値を、r2レジスタにセットし、r0レジスタの値をインクリメント（+1）する（S4）。次に、S5で、a0レジスタに格納されるデータ値を、r2レジスタに格納されるアドレス先

にセットする (S5)。

【0008】

S3～S5の操作を5回繰り返すと、並び替え前のアドレス (0x0000番地) から (0x0004番地) までに格納されたデータが、(0x2000番地) から (0x2004番地) までのアドレス変換テーブルに従って並び替えられ、それらがアドレス (0x1000番地) からアドレス (0x1004番地) までに格納される (図9)。

【0009】

【特許文献1】特開2001-196940

【発明の開示】

【発明が解決しようとする課題】

【0010】

このような従来の並べ替え方法では、多くの処理が必要であり、また、多くのメモリ容量を必要とする問題があった。

【0011】

本発明は、従来のデータ並び替え方法が有する上記問題点に鑑みてなされたものであり、本発明の目的は、処理数 (命令数) を減らして処理時間を短縮し、しかも少ないメモリ容量で実現できる、効率良いデータ並び替え方法を提供することである。

【課題を解決するための手段】

【0012】

上記課題を解決するための本発明のデータ並び替え方法は、a) 第1のデータ格納部にデータを格納するステップと、b) スタックにデータ並び替え情報を格納するステップと、c) 第1のデータ格納部に格納されたデータを読み出し、このデータをスタックに格納されたデータ並び替え情報に基づいて第2のデータ格納部に格納するステップとを有することを特徴とする。

【0013】

このような方法により、処理数を減らして処理時間を短縮し、しかも少ないメモリ容量で、効率良いデータの並び替えが実現できる。また、任意の規則による並べ替えが可能になる。

【0014】

データ並び替え情報は、第2のデータ格納部のアドレスで構成してもよい。また、第1のデータ格納部をレジスタとし、第2のデータ格納部をランダムアクセスメモリとしてもよい。また、並び替えた結果を順次メモリに格納してもよく、DSPなどにより演算処理した結果を順次メモリに格納してもよい。

【0015】

また、アドレス変換テーブルと、これに対応するスタックポインタとを用いて、読み出しと前記格納とを行ってもよい。アドレス変換テーブルと、これに対応するスタックポインタは複数用いてもよく、読み出しと、並び替え結果または演算結果の格納とをパイプライン処理することで、より一層の処理速度の高速化が図れる。

【0016】

また、読み出したアドレスとオフセットレジスタとの論理和 (OR) または加算 (ADD) を演算するステップをさらに含むことで、複数のデータ列の並べ替えが可能になる。特に、使用するアドレス空間を縮小させるためには、論理和演算の利用が有効であり、また、きめ細かくアドレス空間を利用するためには加算演算の利用が有効である。

【0017】

また、スタックポインタの代わりに、並べ替え専用のレジスタを用いることもできる。並べ替え専用のレジスタを用いることにより、任意のポインタ更新が可能になり、より効率の良いデータの並び替えが可能になる。

【発明の効果】

【0018】

以上説明したように、本発明によれば、従来の方法より処理数 (命令数) を減らして処

理時間を短縮して効率よく、しかも少ないメモリ容量で実現でき、任意の並び替え規則に対応できる、データ並び替え方法が提供できた。

【発明を実施するための最良の形態】

【0019】

以下に添付図面を参照しながら、本発明にかかるデータ並び替え方法の好適な実施の形態について詳細に説明する。なお、本明細書及び図面において、実質的に同一の機能構成を有する構成要素については、同一の符号を付することにより重複説明を省略する。

【0020】

(第1の実施の形態)

第1の実施の形態によるデータ並び替え方法は、特定の規則をアドレス変換テーブルに格納する段階と、演算処理のためのデータが格納されるアドレスを、スタックポインタを用いてアドレス変換テーブルから順次読み出す段階とを含む。ここでも説明の簡略化のため、一次元の並び替えについて説明するが、本発明はこれに限定されるものではない。

【0021】

図1は、アドレス変換テーブルとスタックポインタを用いた第1の実施の形態のデータ並び替え方法を説明するための、アドレス変換テーブルの状態とスタックポインタの位置を示す図である。スタックポインタは、スタックの先頭アドレスを保持するためのレジスタで、一般的にはデータをプッシュダウン（格納）するとインクリメント（+1）され、データをポップアップ（読み出し）するとデクリメント（-1）される。スタックでは、最後に格納されたデータが最初に読み出される、LIFO（Last-In-First-Out；後入れ先出し）構造をとる。

【0022】

図1において、「アドレス：0x8000」などはメモリの16進数表示によるアドレスを示す。まず、並び替えの情報は、スタックにてプッシュダウンされ、アドレス変換テーブルとして格納される。例えば、メモリ10のアドレス（0x8000番地）には、アドレス（0x1002番地）が、メモリ10のアドレス（0x8001番地）には、アドレス（0x1000番地）が、順次格納される。

【0023】

DSPにより順次演算処理を行う場合は、スタックをポップアップし、スタックポインタの示すアドレス（0x8004番地）に格納されるアドレス（0x1003番地）が参照され、メモリ10のアドレス（0x1003番地）に格納されるデータが読み出される。DSPによるそのデータの演算処理が終了すると、スタックポインタ30はポップアップによりデクリメントされているので、今度はアドレス（0x8003番地）を示し、ここに格納されるアドレス（0x1001番地）が参照される。以下、同様にして、アドレス変換テーブルが参照される（図1）。

【0024】

図2は、第1の実施の形態の並び替え方法を実現するための流れ図と、プログラム例である。図2の流れ図及びプログラム例は、アドレス変換テーブル（スタック）に予め変換用のアドレスが格納され、スタックポインタが所定のポインタ位置に移動した後の処理について示している。ここでは、1つのアドレス格納用のレジスタr0と、1つのデータ格納用レジスタa0を用いている。プログラム例において、「pop」はスタックの先頭アドレスの値を、右のレジスタにコピーする命令であり、「mov」は前述の通りであり、「（）」付きの文字／数は、そのアドレスやレジスタに格納されるデータの値を、「（）」無しの文字／数は、そのアドレス値やレジスタそのものを示している。

【0025】

まず、S10で、DSPが演算処理を開始し、その結果をa0レジスタに格納する（S10）。次に、S11で、スタックからポップアップして、スタックポインタが示すアドレス変換テーブルのアドレス（0x8004番地）に格納されるアドレス（0x1003番地）を、r0レジスタにセットする（S11）。次に、S12で、a0レジスタに格納された演算結果を、r0レジスタに格納されたアドレス（0x1003番地）に格納する

(S12)。次に、S13で、r0レジスタがアドレス変換テーブル領域からはずれたか否かを判断する(S13)。判断の結果、領域からはずれていたら、スタックポインタを初期領域に戻し、はずれていなかったら次の演算処理に進む(図2)。なお、演算の区切りなどにおいて、スタックポインタを初期状態に戻す場合は、S13、S14の処理は省略可能である。

【0026】

スタックポインタが、アドレス(0x8004番地)からアドレス(0x8000番地)まで進むと、あらかじめスタックにプッシュダウンした順序で演算処理が実行される。

【0027】

以上示したように、スタックポインタを用いることにより、少ない処理数で所定のアドレスにデータを書き込むことが可能になる。また、スタックに任意の並び替え情報を書き込むことができるため、任意の並び替えに対応可能である。このようなことは、ハードウェアのみによる処理では困難である。

【0028】

上記説明では、データの格納(書き込み)に関して説明したが、データの読み出しにおいても同様の方法で実現可能である。この場合、図2のプログラム例は、

```
pop    r0
mov     (r0), a0
```

のようになる。

【0029】

また、図2のプログラム例では、pop命令とmov命令を別々に実行しているが、pop命令とmov命令とを1命令で実行できるシステムにおいては、1命令で実行可能になる。

【0030】

また、図2のプログラム例では、pop命令によってメモリからの読み出しが行われ、mov命令によってメモリへの書き込みが行われているが、メモリへの読み出しと書き込みが別ステージのシステムの場合は、これをパイプライン処理することによって、1命令で実行が可能になる。その例を、図3に示す。

【0031】

図3に示すデータ読み出し時の並べ替えの際に、「pop r0」と「mov (r0), a0」とが1命令の場合は、pop命令においてメモリからの読み出しが行われ、mov命令においてもメモリからの読み出しが行われるため、ステージt3のReadにおいて競合が発生する。但し、競合を回避するシステムである場合は、データ読み出しの場合でも1命令で実行可能である。競合の回避策としては、同時読み出しが発生した場合に、一方の読み出しを優先して実行し、他方の読み出しを遅らせるなどの処理により実現すればよい。また、独立のメモリから読み出しを行う場合は競合が発生しないため、1命令で実行することが可能である。

【0032】

第1の実施の形態では、スタック領域が1つのみの場合について説明したが、複数の並び替え情報を使用する場合は、複数のスタック処理ができるように、スタックポインタを複数用意すればよい。なお、この場合は、スタックを操作するpush命令、pop命令、及び、「スタックポインタを初期化」するためのスタックポインタへのmov命令は、複数用意することになる。

【0033】

(第2の実施の形態)

第2の実施の形態によるデータ並び替え方法では、第1の実施の形態による方法に加えて、読み出したアドレスとオフセットレジスタとの論理和(OR)または加算(ADD)を演算する段階をさらに加える。

【0034】

オフセットレジスタを設けて、スタックには並び替えアドレスの下位ビットを格納し、

読み出したアドレスとオフセットレジスタとの論理和 (OR) または加算 (ADD) を演算して並び替えのアドレスを生成することで、複数のデータ列の並び替えを可能にする。

【0035】

ここで、「OR」演算は、ビット毎の「or」演算を意味し、「ADD」演算は、「加算」演算を意味する。「OR」または「ADD」の選択は、例えば制御レジスタを設けて、任意に制御するようにすればよい。

【0036】

第2の実施の形態によれば、第1の実施の形態による効果に加え、複数のデータ列の並び替えが可能になる。

【0037】

(第3の実施の形態)

第3の実施の形態によるデータ並び替え方法では、第1の実施の形態におけるスタックポインタの代わりに、並び替え専用のレジスタを用いる。並び替え専用のレジスタを用いることにより、任意のポインタ更新が可能になり、より効率の良いデータの並び替えが可能になる。

【0038】

図4は、アドレス変換テーブルと並び替え専用レジスタ (rr) を用いた第3の実施の形態のデータ並び替え方法を説明するための、アドレス変換テーブルの状態と、並び替え専用レジスタ (rr) の内容を示す図である。

【0039】

図5は、第3の実施の形態の並び替え方法を実現するための流れ図と、プログラム例である。図5の流れ図及びプログラム例は、アドレス変換テーブルに予め変換用のアドレスが格納され、並び替え専用レジスタ rr には所定のポインタ初期値が格納された状態後の処理について示している。ここでは並び替え専用レジスタ rr の他に、1つのアドレス格納用のレジスタ r0 と、1つのデータ格納用レジスタ a0 を用いている。プログラム例において、「mov」などは前述の通りである。

【0040】

まず、S30で、DSPが演算処理を開始し、その結果をa0レジスタに格納する (S30)。次に、S31で、並び替え専用レジスタ rr に格納されたアドレス値が示すアドレス変換テーブルのアドレス (0x8004番地) に格納されるアドレス (0x0003番地) を読み出し、r0レジスタにセットし、並び替え専用レジスタ rr をデクリメント (-1) する (S31)。次に、S32で、a0レジスタに格納された演算結果を、r0レジスタに格納されたアドレス (0x0003番地) に格納する (S32)。次に、S33で、r0レジスタがアドレス変換テーブル領域からはずれたか否かを判断する (S33)。判断の結果、領域からはずれていたら、並び替え専用レジスタ rr を初期状態に戻し、はずれていなかったら次の演算処理に進む (図5)。

【0041】

並び替え専用レジスタ rr の内容が、アドレス (0x8004番地) からアドレス (0x8000番地) まで変化すると、あらかじめメモリに格納された順序どおりに演算処理が実行される。

【0042】

第3の実施の形態によれば、第1の実施の形態による効果と同様の効果が得られる。並び替え専用レジスタ rr からデータを読み出した際に、並び替え用レジスタ rr を-1する説明をしたが、用途に応じて任意のmodifyを可能とする。例えば、modifyとしては、任意の+または-、モジュールアドレッシングなどが挙げられる。modifyは、一般的なアドレスレジスタに備わる全てのアドレッシングに関してサポートするものを用いることができる。モジュールアドレッシングの場合は、並び替え専用レジスタと同様に、モジュール幅などを指定するためのレジスタを専用を用意し、その他のアドレッシングについても必要なレジスタを用意すればよい。このように、モジュールアドレッシングを用いることにより、定期的なポインタの再設定が不要となり、効率的な並び替えが

可能となる。

【0 0 4 3】

(第4の実施の形態)

第4の実施の形態によるデータ並び替え方法は、アドレス変換テーブルに格納するデータに、バイトライト情報を含める。図6は、第4の実施の形態における、アドレス変換テーブルの状態、スタックポインタの位置、制御レジスタ40とオフセットレジスタ50の機能を示す図である。第1～第3の実施の形態では、データの書き込みを固定のビット長(例えば、32bitなど)に対して行っている。しかし、演算処理によっては、32bit中の任意のbit(例えば、8bitなど)に対してのみ、書き込みを行う必要がある。

【0 0 4 4】

第4の実施の形態では、アドレス変換テーブルに並び替え情報を格納する際に、下位のビットをバイトライト制御bitにすることにより、任意の場所に書き込むことを可能にする。

【0 0 4 5】

例えば、表1、図7に示すように、並び替え情報の下位2bitを、バイトライト情報とする。

【0 0 4 6】

【表1】

バイトライト情報	意味
0 0	bit 7～0に対するWrite
0 1	bit 15～8に対するWrite
1 0	bit 23～16に対するWrite
1 1	bit 31～24に対するWrite

【0 0 4 7】

さらに、バイトライト制御レジスタを設けて、バイトライト可の場合は、アドレスの下位2bitをバイトライト情報としてメモリに供給する。残りの上位14bitは、アドレスとしてメモリに供給する(図7参照)。

【0 0 4 8】

このように、第4の実施の形態によれば、並び替え情報の下位bitをバイトライト情報とすることにより、所定のbitに対してのみ書き込みが可能になる。上記の例では、32bitを8bitずつに分割した例であるが、分割の仕方に関しては限定されない。分割の際、バイトライト情報のbit数は、必要bit数とすればよい。また、バイトライト制御レジスタを設置することにより、並び替えのために使用しているメモリを、一般的な用途のために使用することも可能になる。

【0 0 4 9】

以上、添付図面を参照しながら本発明にかかるデータ並び替え方法の好適な実施形態について説明したが、本発明はかかる例に限定されない。当業者であれば、特許請求の範囲に記載された技術的思想の範疇内において各種の変更例または修正例に想到し得ることは明らかであり、それらについても当然に本発明の技術的範囲に属するものと了解される。

【0 0 5 0】

例えば、多次元の並べ替えについても適用できる。

【産業上の利用可能性】

【0 0 5 1】

本発明は、デジタル通信やデジタル信号処理におけるデータ処理方法に関し、特に、データ並び替え方法に利用可能である。

【図面の簡単な説明】

【 0 0 5 2 】

【図 1】第 1 の実施の形態における、アドレス変換テーブルの状態とスタックポインタの位置を示す説明図である。

【図 2】第 1 の実施の形態の並べ替え方法を実現するための流れ図と、プログラム例を示す説明図である。

【図 3】第 1 の実施の形態における、パイプライン処理する場合のステージ表を示す説明図である。

【図 4】第 3 の実施の形態における、アドレス変換テーブルの状態と並び替え専用レジスタ (r r) の内容を示す説明図である。

【図 5】第 3 の実施の形態の並べ替え方法を実現するための流れ図と、プログラム例を示す説明図である。

【図 6】第 4 の実施の形態における、アドレス変換テーブルの状態、スタックポインタの位置、制御レジスタ 4 0 とオフセットレジスタ 5 0 の機能を示す説明図である。

【図 7】第 4 の実施の形態における、バイトライト信号を説明する説明図である。

【図 8】従来のデータ並び替え方法を説明するための、データ並び替え前後のデータ配置状態と、アドレス変換テーブルの状態を示す説明図である。

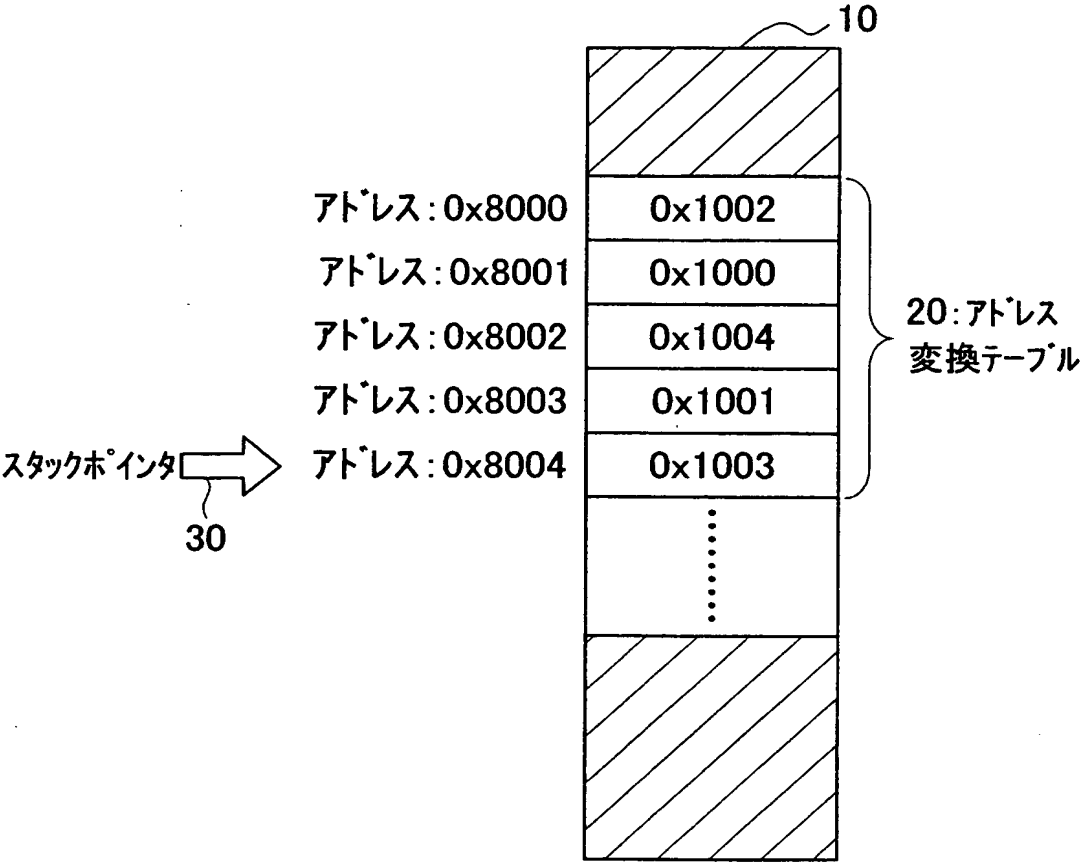
【図 9】従来の並べ替え方法を実現するための流れ図と、プログラム例を示す説明図である。

【符号の説明】

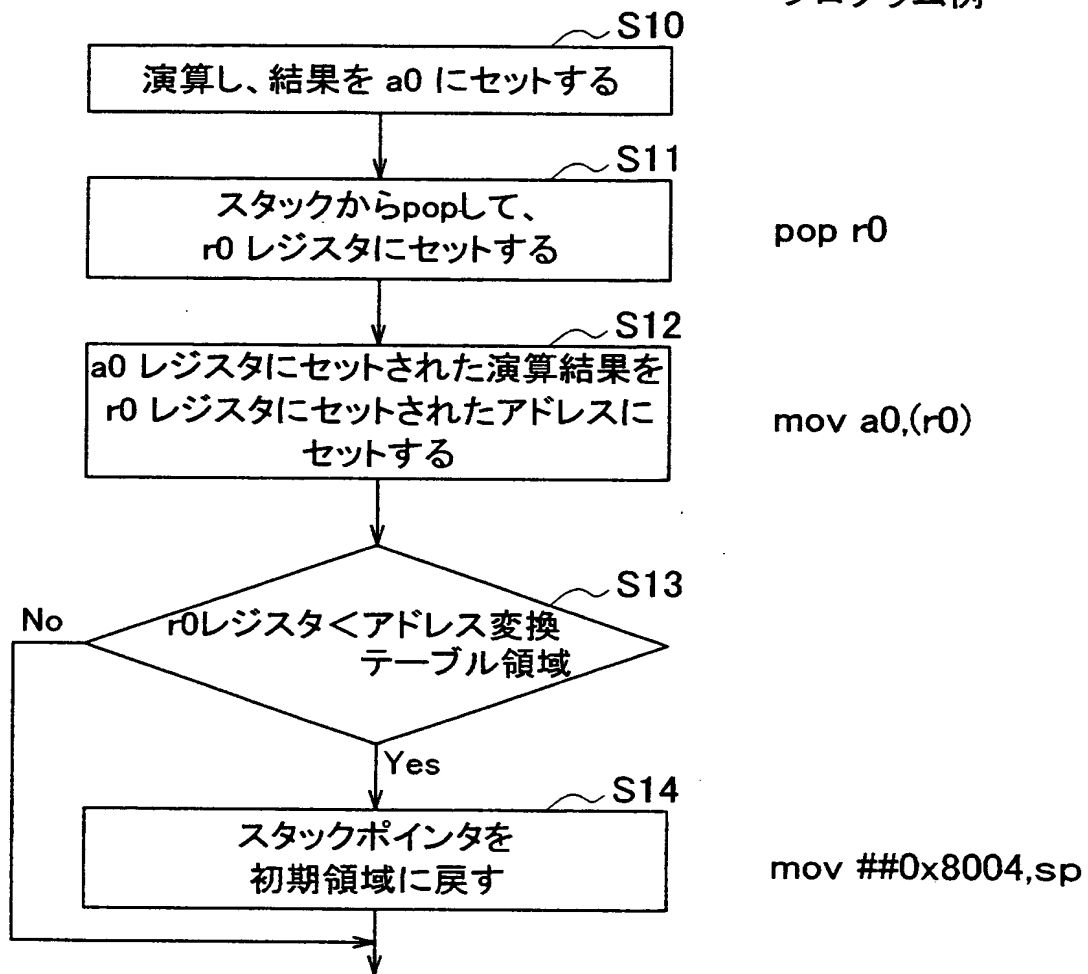
【 0 0 5 3 】

- 1 0 メモリ
- 2 0 アドレス変換テーブル
- 3 0 スタックポインタ
- r r 並び替え専用レジスタ
- 4 0 制御レジスタ
- 5 0 オフセットレジスタ

【書類名】 図面
【図 1】



【図 2】

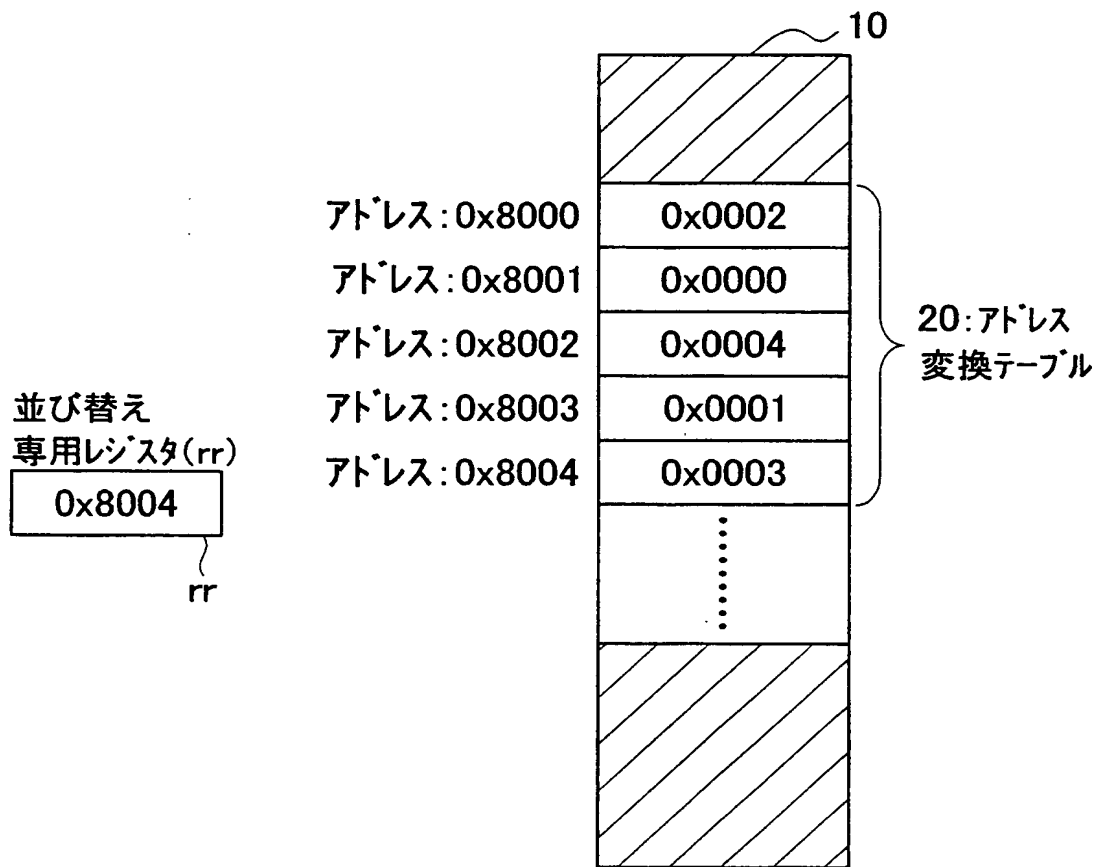


【図 3】

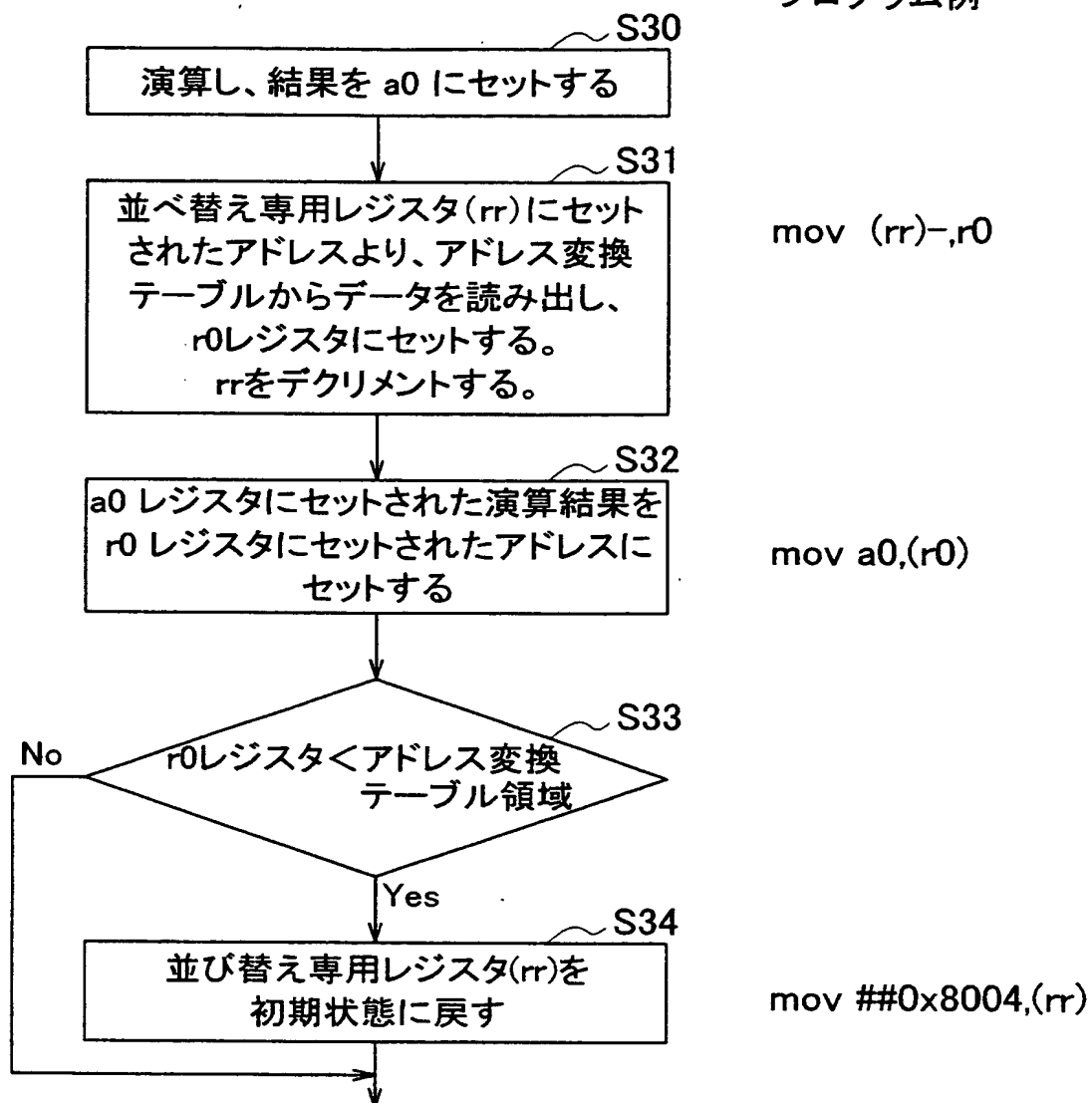
パイプライン ステージ	t1	t2	t3	t4	t5	t6
Fetch	n1	n2	n3	n4		
Decode		n1	n2	n3	n4	
Read			n1 (popによりメモリから読み出し)	n2	n3	n4
Execute				n1	n2	n3
Write					n1 (movによりメモリへ書き込み)	n2

‘pop rol mov a0,(r0)’をn1命令とする

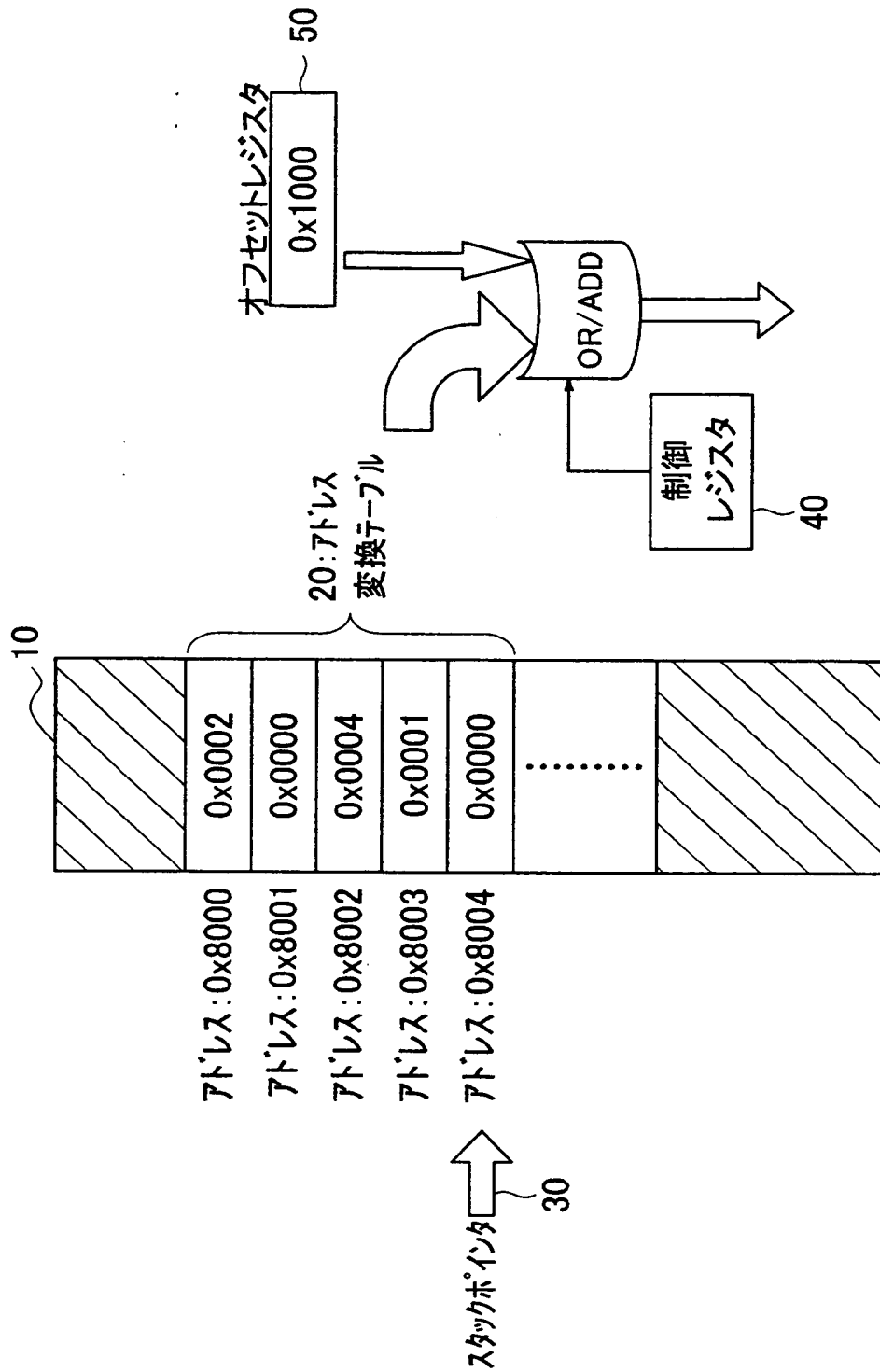
【図 4】



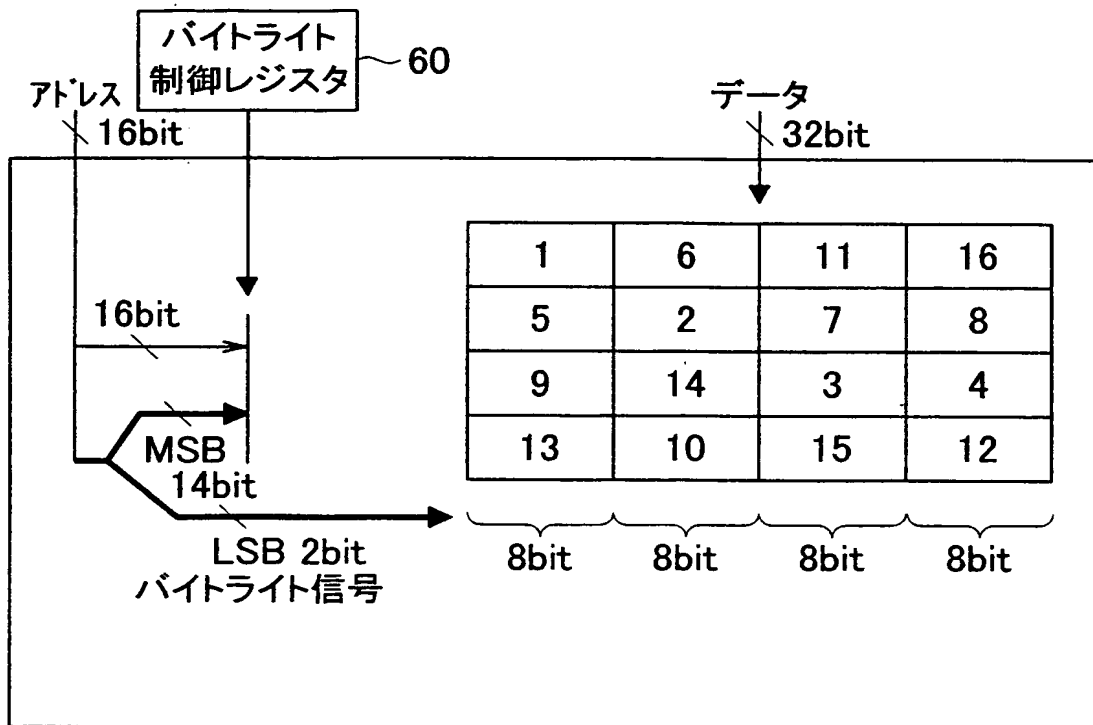
【図 5】



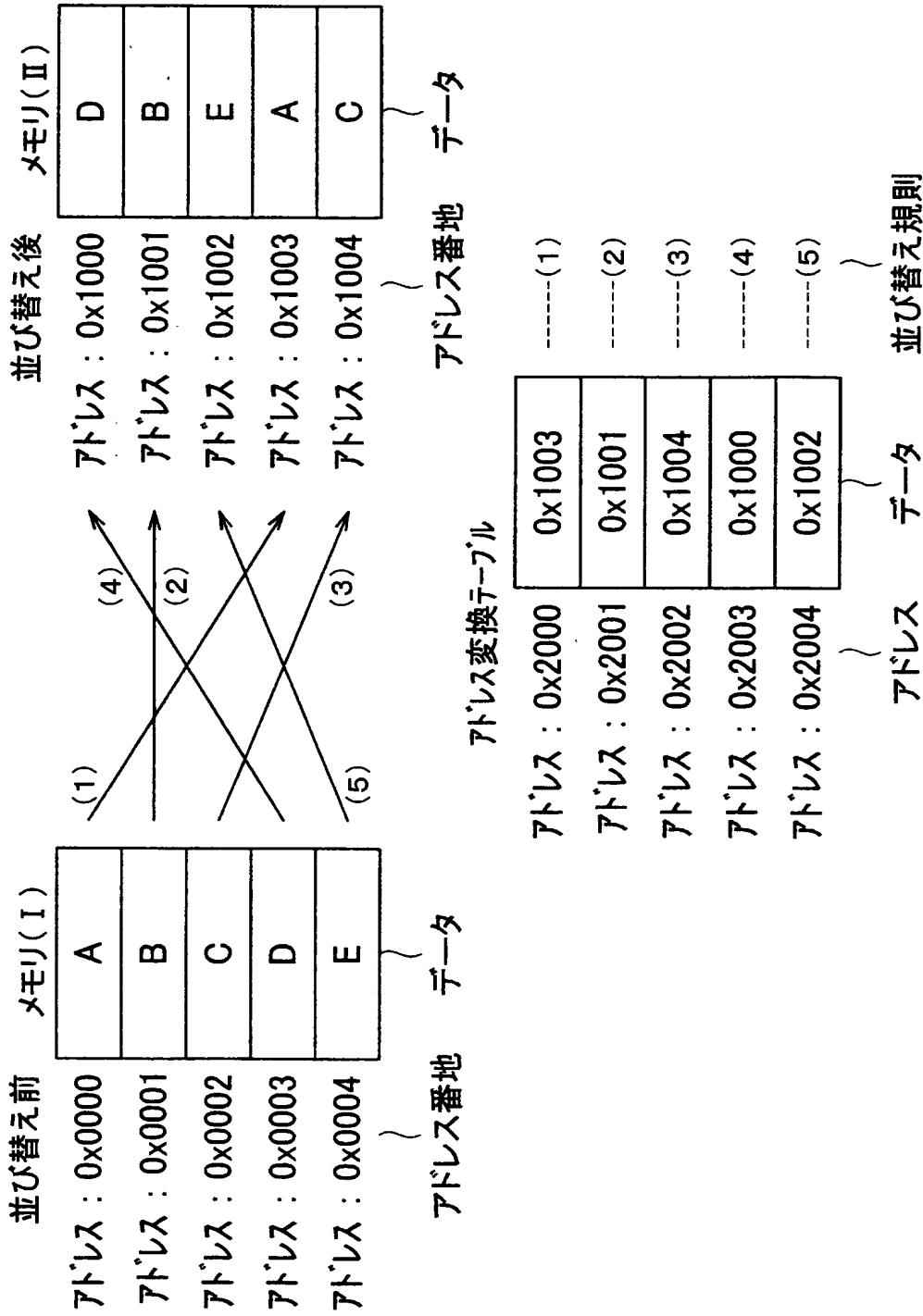
【図 6】



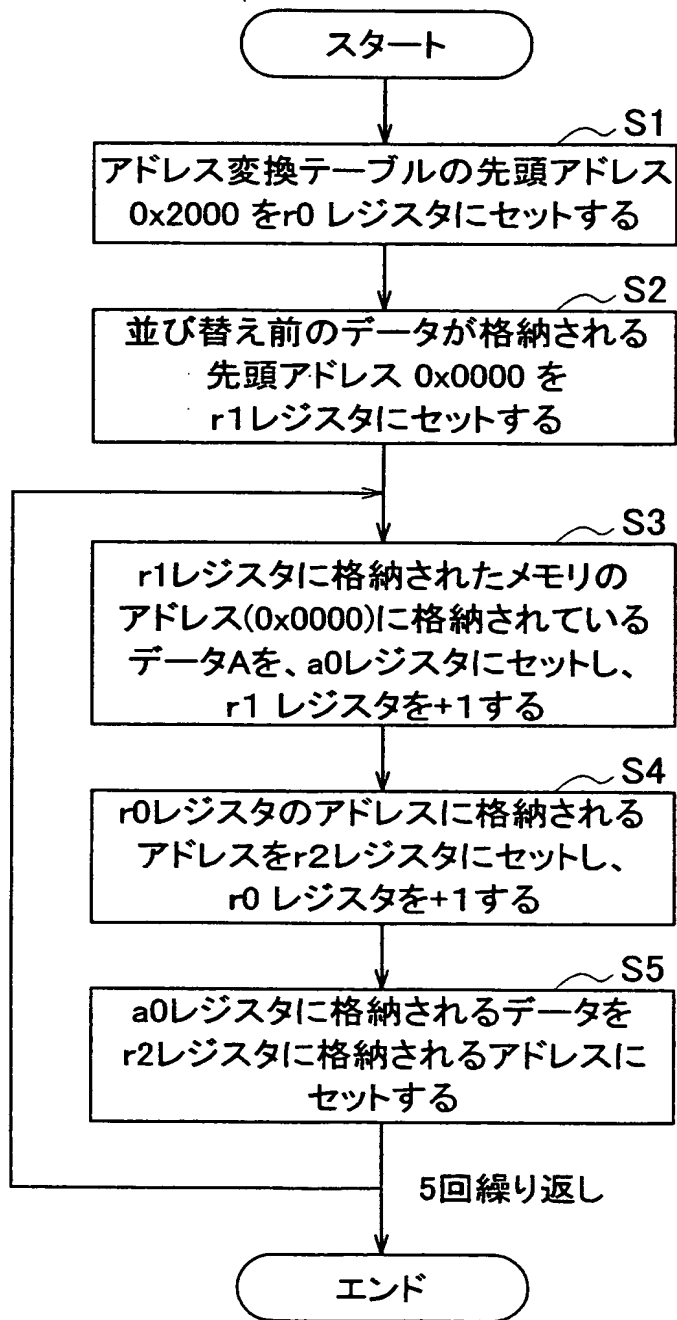
【図 7】



【図 8】



【図 9】



プログラム例

```
mov 0x2000,r0
```

```
mov 0x0000,r1
```

```
mov (r1)+,a0
```

```
mov (r0)+,r2
```

```
mov a0, (r2)
```

【書類名】 要約書

【要約】

【課題】 処理数を減らして処理時間を短縮した，効率良い，任意の規則に対応するデータ並び替え方法を提供する。

【解決手段】 本発明のデータ並び替え方法は， a) 第 1 のデータ格納部にデータを格納するステップと， b) スタックにデータ並び替え情報を格納するステップと， c) 第 1 のデータ格納部に格納されたデータを読み出し，このデータをスタックに格納されたデータ並び替え情報に基づいて第 2 のデータ格納部に格納するステップとを有することを特徴とする。

【選択図】 図 1

特願 2 0 0 3 - 3 1 8 4 9 3

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 0 2 9 5]

1. 変更年月日

1 9 9 0 年 8 月 2 2 日

[変更理由]

新規登録

住 所

東京都港区虎ノ門 1 丁目 7 番 1 2 号

氏 名

沖電気工業株式会社